

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-200214

(43)Date of publication of application : 18.07.2000

(51)Int.Cl.

G06F 12/06
G11C 11/401

(21)Application number : 11-226819

(71)Applicant : LUCENT TECHNOL INC

(22)Date of filing : 10.08.1999

(72)Inventor : RICHARD JOSEPH NIISHIAA

(30)Priority

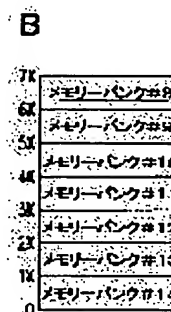
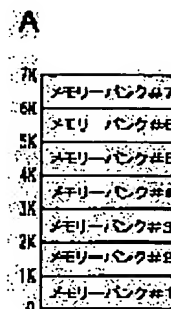
Priority number : 98 141511 Priority date : 27.08.1998 Priority country : US

(54) MEMORY SYSTEM AND ASSIGNING METHOD FOR MEMORY BANKS

(57)Abstract:

PROBLEM TO BE SOLVED: To make adjustable changing demands of access agents by allocating 1st parts of memory banks to a 1st memory map of a 1st process agent and 2nd parts to a 2nd memory map of a 2nd process agent.

SOLUTION: The 1st memory bank is assigned to the memory map of the 1st agent preferably in order as shown in Fig.A. Consequently, if the memory system is so initially sectioned into seven memory banks that the 1st agent can access and seven memory banks that the 2nd agent can access, the starting seven memory banks are assigned in the increasing order to the memory map of the 1st agent. Memory banks for order agents are assigned in the decreasing order (top to bottom) to the memory map of the 2nd agent from the bottom to the top.



LEGAL STATUS

[Date of request for examination] 23.01.2001

[Date of sending the examiner's decision of rejection] 06.09.2004

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection] 2004-24909

[Date of requesting appeal against examiner's decision of rejection] 06.12.2004

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2000-200214
(P2000-200214A)

(43) 公開日 平成12年7月18日 (2000.7.18)

(51) Int.Cl.	識別記号	F I	テーマコード(参考)
G 0 6 F 12/06	5 1 5	G 0 6 F 12/06	5 1 5 H
G 1 1 C 11/401		G 1 1 C 11/34	3 6 2 H

審査請求 未請求 請求項の数27 O L 外国語出願 (全 39 頁)

(21) 出願番号 特願平11-226819

(22) 出願日 平成11年8月10日 (1999.8.10)

(31) 優先権主張番号 09/141511

(32) 優先日 平成10年8月27日 (1998.8.27)

(33) 優先権主張国 米国 (U S)

(71) 出願人 596077259

ルーセント テクノロジーズ インコーポ
レイテッド

Lucent Technologies
Inc.

アメリカ合衆国 07974 ニュージャージー
ー、マレーヒル、マウンテン アベニュー
600-700

(74) 代理人 100081053

弁理士 三俣 弘文

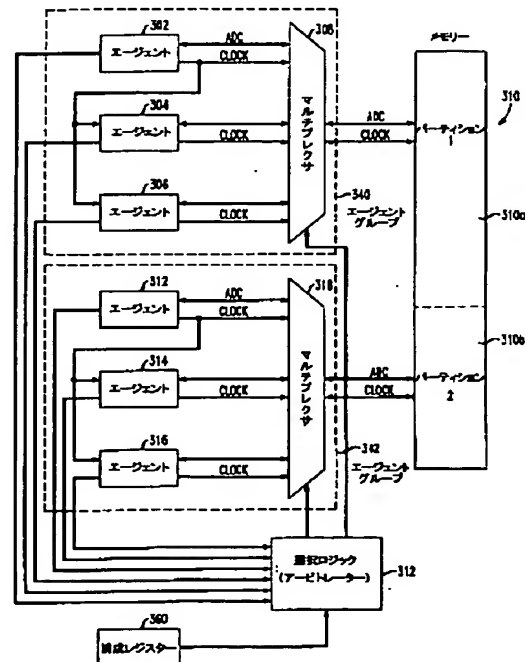
最終頁に続く

(54) 【発明の名称】 メモリーシステムおよびメモリーバンクを割り当てる方法

(57) 【要約】

【課題】 複数のアクセスエージェントの変化する需要を調整するようなシングルポート型共有メモリーのメモリーシステム。

【解決手段】 全体で最低アドレスアブル端および最高アドレスアブル端を有する複数の隣接メモリーバンクからなる。複数の隣接メモリーバンクそれぞれをスイッチングするようにスイッチが構成し、第1処理エージェントと第2エージェントの1つからアクセスされる。制御回路は、複数の隣接メモリーバンクの第1のサブを、最低アドレスアブル端から第1処理エージェントの第1メモリーマップへと昇順で割り当て、複数の隣接メモリーバンクの第2サブを、最高アドレスアブル端から第2処理エージェントの第2メモリーマップへと降順で割り当てる。



【特許請求の範囲】

【請求項 1】 (A) 全体で最低アドレスابل端と最高アドレスابل端を有する複数のメモリーバンクと、

(B) 最低アドレスابل端から昇順で前記複数のメモリーバンクの第 1 部分を第 1 処理エージェントの第 1 メモリーマップへと割り当て、最高アドレスابل端から降順で前記複数のメモリーバンクの第 2 部分を第 2 処理エージェントの第 2 メモリーマップへと割り当てる制御モジュールとを有することを特徴とするメモリーシステム。

【請求項 2】 前記複数のメモリーバンクは、隣接していることを特徴とする請求項 1 記載のメモリーシステム。

【請求項 3】 (C) 第 1 処理エージェントと第 2 処理エージェントの内の 1 つからアクセスされるため前記複数のメモリーバンクのそれぞれをスイッチングするように構成したスイッチを更に有することを特徴とする請求項 1 記載のメモリーシステム。

【請求項 4】 前記制御モジュール (B) は、前記複数のメモリーバンクの第 1 部分と前記複数のメモリーバンクの第 2 部分を規定する構成レジスターを有することを特徴とする請求項 1 記載のメモリーシステム。

【請求項 5】 前記複数のメモリーバンクは、何れの時点においても第 1 処理エージェントと第 2 処理エージェントの内の一方のみに割り当てることを特徴とする請求項 1 記載のメモリーシステム。

【請求項 6】 前記複数のメモリーバンクは、同期メモリーバンクであることを特徴とする請求項 1 記載のメモリーシステム。

【請求項 7】 前記複数のメモリーバンクは、非同期メモリーバンクであることを特徴とする請求項 1 記載のメモリーシステム。

【請求項 8】 複数の処理エージェントの間で共有するために複数のメモリーバンクを割り当てる方法であって、

(A) 第 1 処理エージェントの低いアドレスが当該複数のメモリーバンクの最後の 1 つと関連づけられ、第 1 処理エージェントの高いアドレスが当該複数のメモリーバンクの最後の 1 つと関連づけられるように複数のメモリーバンクへと第 1 処理エージェントにアクセスさせるステップと (B) 第 2 処理エージェントの低いアドレスが当該複数のメモリーバンクの前記最後の 1 つに関連づけられ、第 2 処理エージェントの高いアドレスが当該複数のメモリーバンクの最後の 1 つと関連づけられるように当該複数のメモリーバンクへと第 2 処理エージェントにアクセスさせるステップとを有することを特徴とする方法。

【請求項 9】 当該複数の処理エージェントの前記最初の 1 つは、第 1 処理エージェントのメモリーマップにおけるアドレス 0 と関連づけられることを特徴とする請求項 8 記載の方法。

【請求項 10】 当該複数の処理エージェントの前記最初の 1 つは、第 2 処理エージェントのメモリーマップにおけるアドレス 0 と関連づけられることを特徴とする請求項 8 記載の方法。

【請求項 11】 当該複数の処理エージェントの前記最初の 1 つは、第 2 処理エージェントのメモリーマップにおけるアドレス 0 と関連づけられることを特徴とする請求項 9 記載の方法。

【請求項 12】 当該複数のメモリーバンクは、同期メモリーであることを特徴とする請求項 9 記載の方法。

【請求項 13】 当該複数のメモリーバンクは、非同期メモリーであることを特徴とする請求項 9 記載の方法。

【請求項 14】 当該複数のメモリーバンクのそれぞれは、第 1 処理エージェントと第 2 処理エージェントの両方によって同時にアクセスすることができないことを特徴とする請求項 9 記載の方法。

【請求項 15】 2 のエージェントに複数の共有メモリーバンクを割り当てる方法であって、

(A) 複数の共有メモリーバンクを昇順で第 1 エージェントのメモリーマップへと割り当てるステップと、

(B) 当該複数の共有メモリーバンクを降順で第 2 エージェントのメモリーマップへと割り当てるステップとを有することを特徴とする方法。

【請求項 16】 当該複数の共有メモリーバンクは、同期メモリーバンクであることを有することを特徴とする請求項 15 記載の方法。

【請求項 17】 2 のエージェントに複数の共有メモリーバンクを割り当てる方法であって、

(A) 2 の処理エージェントそれぞれに対して逆順で複数の共有メモリーバンクを割り当てるステップを有することを特徴とする方法。

【請求項 18】 複数の処理エージェントの間で共有するために複数のメモリーバンクを割り当てる装置であって、

(A) 第 1 処理エージェントの低いアドレスが当該複数のメモリーバンクの最後の 1 つと関連づけられ、第 1 処理エージェントの高いアドレスが当該複数のメモリーバンクの最後の 1 つと関連づけられるように複数のメモリーバンクへ第 1 処理エージェントにアクセスさせる手段と、

(B) 第 2 処理エージェントの低いアドレスが当該複数のメモリーバンクの前記最後の 1 つと関連づけられ、第 2 処理エージェントの高いアドレスが当該複数のメモリーバンクの最初の 1 つと関連づけられるように当該複数のメモリーバンクへ第 2 処理エージェントにアクセスさせる手段とを有することを特徴とする装置。

【請求項 19】 当該複数の処理エージェントの前記最初の 1 つは、第 1 処理エージェントのメモリーマップにおけるアドレス 0 と関連づけられることを特徴とする請求項 18 記載の装置。

【請求項 20】 当該複数の処理エージェントの前記最初の 1 つは、第 2 処理エージェントのメモリーマップにおけるアドレス 0 と関連づけられることを特徴とする請求項 18 記載の装置。

【請求項 21】 当該複数の処理エージェントの前記最初の 1 つは、第 2 処理エージェントのメモリーマップにおけるアドレス 0 と関連づけられることを特徴とする請求項 20 記載の装置。

【請求項 22】 当該メモリーバンクは、同期メモリーであることを特徴とする請求項 18 記載の装置。

【請求項 23】 2 のエージェントに複数の共有メモリーバンクを割り当てる装置であって、

(A) 複数の共有メモリーバンクを昇順で第 1 エージェントのメモリーマップへと割り当てる手段と、

(B) 当該複数の共有メモリーバンクを降順で第 2 エージェントのメモリーマップへと割り当てる手段とを有することを特徴とする装置。

【請求項 24】 当該複数の共有メモリーバンクは、同期メモリーバンクであることを有することを特徴とする請求項 23 記載の装置。

【請求項 25】 2 のエージェントに複数の共有メモリーバンクを割り当てる装置であって、

(A) 2 の処理エージェントそれぞれに対して逆順で複数の共有メモリーバンクを割り当てる手段を有することを特徴とする装置。

【請求項 26】 当該複数の共有メモリーバンクは、同期メモリーからなることを特徴とする請求項 25 記載の装置。

【請求項 27】 (A) 第 1 処理エージェントと、

(B) 第 2 処理エージェントと、

(C) 第 1 処理エージェントおよび第 2 処理エージェントがアクセスできる共有同期メモリーと

(D) 最低アドレスアブル端から昇順で前記複数のメモリーバンクの第 1 部分を第 1 処理エージェントの第 1 メモリーマップへと割り当て、第 2 処理エージェントの第 2 メモリーマップへと最高アドレスアブル端から降順で前記複数の共有同期メモリーを割り当てる制御モジュールとからなるコンピューター装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、複数の処理エージェントによるメモリーの共有に関し、特に、複数の処理エージェントの間のシングルポート型共有メモリーの効率的で柔軟性のあるパーティショニングに関する。

【0002】

【従来の技術】 今日のプロセッサの速度が今までかつてないほど速くなっているため、このような速度要件を満たすようなメモリー設計が試みられてきた。シングルシステムにおいて複数の特定のタスク指向プロセッサの集積が増しているため、高速メモリーの有効利用が設

計上の課題になっている。

【0003】 例えば、高速メモリー速度を達成するため、広い用途のために SDRAM 技術が開発され、DRAM や SRAM のような非同期型メモリーのアクセス時間と高速プロセッサのニーズとの間のギャップを狭くしてきた。同期メモリー (例えば、SDRAM 技術) は、ファースト (fast) DRAM における業界の進歩を高速インターフェースと組み合わせている。

【0004】 機能的に SDRAM は従来の DRAM と似ている。即ち、ダイナミックでありリフレッシュすることを必要とする。しかし、SDRAM のアーキテクチャーは標準的な DRAM よりも優れている。例えば、SDRAM はスループットを改善するため内部パイプラインを用い、出力データのギャップをなくするため、別々のメモリーバンクの間でオンチップインターリーブを用いる。

【0005】 SDRAM を同期させて (DRAM を非同期に用いるのに対して) 用いる考えは、ハイエンドプロセッサのデータ転送の需要が増加しているため発生した。SDRAM 回路設計は、従来の非同期的メモリーデバイスのようにレベル/パルス幅駆動されるのではなく、状態マシン動作に基づく。また、同期的メモリーアクセス技術はシステムノイズに対するマージンを改善する。なぜなら入力がレベル駆動されていないからである。代わりに、入力はシステムクロックによりラッチされる。全てのタイミングが同じ同期クロックに基づくので、仕様上のマージンが良くなる。また、SDRAM アクセスはプログラマブルなので、バスの利用を改善することができる。なぜなら、プロセッサを SDRAM 出力と同期することができるからである。

【0006】 SDRAM のデバイスのコアは、重要な同期制御ロジックを有する標準的な DRAM である。1 つのクロック信号で全てのアドレス、データ、制御信号を同期することにより、SDRAM 技術はパフォーマンスを改善し、設計を単純化し、高速なデータ伝送を与えることができる。

【0007】 同期メモリーは、アクセスしているエージェントからクロック信号を必要とし、そのアクセスしているエージェントの完全な同期操作を可能とさせる。もし複数のエージェントが共有同期メモリーにアクセスを許される場合、各エージェントはその自身のクロック信号をその同期メモリーへと従来の方法で供給しなければならない。しかし、別々のエージェントからのクロック信号は従来技術的に同期してはいない。即ち、他方とインターフェースではない。従って、同期メモリーが 1 つのクロック信号の利用から別のクロック信号へシフトする際、新しいエージェントのクロック信号を同期メモリーにアクセスするのに用いる前に遅延待ち状態を加えなければならない。

【0008】 バースト入出力を与える能力を有する同期

メモリーデバイスがある。これは特に、システム周波数でのキャッシュメモリーの重点の最適化に適する。プログラマブルバーストモードやバースト長さのような発展した特徴により、メモリーシステムのパフォーマンスおよび伝統的な同期メモリーにおける柔軟性を改善させ、バーストにおける個々のアクセスの間の待ち状態（例えば、休止(dormant)クロックサイクル）を挿入する必要性をなくした。

【0009】伝統的なSDRAMデバイスは、独立した固定メモリーセクションを有し、これはここにありいは10 インタリープ方式によりアクセスされることができる。例えば、SDRAMデバイスにおける2つの独立バンクは、同時にそのデバイスが2つの異なる行がアクティブであることを可能にする。このことは、一方のバンクがプリチャージ中に他方のバンクを読んだり書いたりできることを意味する。行をプリチャージしたりアクティブエイトしたりする際のセットアップは、バンクアクセスをインタリープすることにより隠すことができる。

【0010】従来において、タスク指向処理エージェントはそれぞれ、図6に示すような専用メモリーバンクシステムを備える。図6において、2つの別々の処理エー10 ジェントシステム600a、600bはそれぞれ、処理エージェント602、604、専用メモリーバンクシステム508a、508bを有する。各処理エージェント602、604は、マイクロプロセッサ、マイクロコントローラー、デジタルシグナルプロセッサ(DSP)のような適切なプロセッサを用いることができる。

【0011】1つの処理エージェントが、別の処理エージェントから大量の情報を必要とするようなアプリケーションでは、第2処理エージェントが第1処理エー10 ジェントとともに情報を共同処理(co-process)するための非同期メモリーバンクシステム間の情報の伝送の必要性を減らすため、図7に示したようにアービトレーターを用いる従来の非同期メモリーの共有する技術が開発された。

【0012】図7においてシングルポート型コモンメモリーバンクシステム508は2つの処理エージェント502、504によってアクセスされる。マルチプレクサ(MUX)510は、選択された処理エージェント502、504のアドレス、データ、制御バス(ADC)をアービター(arbiter)、即ちアービトレーター512の10 制御の下でシングルポート型コモンメモリーバンクシステム508へとわたす。

【0013】いかなる時間においてもシングルポート型コモンメモリーバンクシステム508にただ1つのエージェントしかアクセスできないように調整(アービレート)されない限り、別々のエージェント502、504によるメモリーのアクセスはクラッシュしてしまう。従って、選択ロジック(アービトレーター512)が従

来マルチプレクサ510を制御するため備えていた。このマルチプレクサ510は、適切なADC信号をシングルポート型コモンメモリーバンクシステム508へと提供する。処理エージェント502、504は通常、シングルポート型コモンメモリーバンクシステム508への優先度をつけられたアクセスのためにアービトレーター512において階層(パイアラキー)を割り当てられ、シングルポート型コモンメモリーバンクシステム508へのアクセスが終わるまでアービトレーター512は他のエージェントによるアクセスを阻止する。

【0014】シングルポート型コモン非同期メモリーを共有する別の従来技術は、図8に示すようなメモリー管理アドレストランスレーター706を用いる。

【0015】図8において、2つの処理エージェント702、704がメモリー管理アドレストランスレーター706およびMUX708の制御の下でシングルポート型コモンメモリーバンクシステム508をアクセスしている。アドレス翻訳機能は、処理エージェント702、704の1つによるメモリーアクセスを所定のメモリー量(例えば、図8に示すように7k)へと不可視にインデックス付け(indexing)するように機能する。

【0016】従って例えば、第1の処理エージェント702によるアドレス0へのメモリーアクセスは、メモリー管理アドレストランスレーター706によってシングルポート型コモンメモリーバンクシステム508における物理アドレス0へとインデックス付けされ、第2処理エージェント704によるアドレス0へのメモリーアクセスは、メモリー管理アドレストランスレーター706によってシングルポート型コモンメモリーバンクシステム508におけるオフセット物理アドレス(例えば、7K)へとインデックス付けされる。

【0017】従来、アドレス翻訳は共有メモリーシステムにおいて固定され、特にエージェントが動作中の時において、処理エージェントのメモリーニーズが変わっても柔軟性を提供してなかった。

【0018】

【発明が解決しようとする課題】このように、複数のアクセスエージェントの変化する需要を調整するようなメモリーシステムの必要性がある。

【0019】

【課題を解決するための手段】本発明の原理に従うメモリーシステムは、集合的に(複数の隣接メモリーバンク全体で)最低アドレスアブル(アドレス割付可能)端および最高アドレスアブル端を有する複数の隣接(contiguous)メモリーバンクからなる。複数の隣接メモリーバンクそれぞれをスイッチングするようにスイッチが構成し、第1処理エージェントと第2エージェントの1つからアクセスされる。制御回路は、複数の隣接メモリーバンクの第1のサブを、最低アドレスアブル端から第1処理エージェントの第1メモリーマップへと昇順で割り当て、複

数の隣接メモリーバンクの第2サブを、最高アドレスから第2処理エージェントの第2メモリーマップへと降順で割り当てる。

【0020】本発明の原理に従い、複数の処理エージェントの間で共有するために複数のメモリーバンクを割り当てる方法は、第1処理エージェントのより低いアドレスが複数のメモリーバンクの最後の1つと関連づけられ、第1処理エージェントのより高いアドレスが複数のメモリーバンクの最後の1つと関連づけられるように複数のメモリーバンクへのアクセスを第1処理エージェントに提供する。第2処理エージェントは、第2処理エージェントのより低いアドレスが複数のメモリーバンクの最後の1つと関連づけられ、第2処理エージェントのより高いアドレスが複数のメモリーバンクの最初の1つと関連づけられるように複数のメモリーバンクへのアクセスを提供される。

【0021】本発明の別の原理に従うと、複数の共有メモリーバンクを2のエージェントへ割り当てる方法は、第1エージェントのメモリーマップに昇順で複数の共有のメモリーバンクを割り当てるステップからなる。複数の共有メモリーバンクは、第2エージェントのメモリーマップに降順で割り当てられる。

【0022】更に別の原理に従うと、複数の共有メモリーバンクを2のエージェントへ割り当てる方法は、2の処理エージェントそれぞれに対して逆順 (opposing) で複数の共有メモリーバンクを割り当てるステップからなる。

【0023】

【発明の実施の形態】図1、図2は、2の別々のエージェントないし処理エージェントグループによってアクセスできる複数のメモリーブロックを備えたシングルポート型メモリーシステムを示してある。本発明は、複数の共有メモリーブロックを2のエージェントの間での割り当てに関する。

【0024】図1において、コモンメモリーシステム310における所定数のメモリーバンクをアクセシングエージェントないしエージェントグループそれぞれに割り当てるように区分された共有メモリーの構成を示している。図1において同期メモリー310を示しているが、本発明は非同期共有メモリー (DRAM、SRAMのような伝統的なメモリー) をも用いることができる。

【0025】図1において、2つのグループのアクセシングエージェントと関連して示してある。即ち、処理エージェント302〜306を含む第1グループの共有メモリーアクセシングエージェント340と、処理エージェント312〜316を含む第2グループの共有メモリーアクセシングエージェント342である。エージェントグループ340、342は同じ数のエージェントを備えるようにしてあるが、本発明は任意の数のエージェントを備えるアクセシングエージェントグループにも同様

に用いることができる。

【0026】メモリーブロック310は、パーティション310a、310bに区分されている。この実施例では2つのパーティション310a、310bで示しているが、本発明は任意の数のパーティションにて用いることができる。パーティションの数はエージェントグループの数と等しいことが好ましい。

【0027】別々のエージェントグループ340、342は共有メモリー310の対応する区分された部分にアクセスする。パーティション310a、310bは、いかなる大きさでもよく構成レジスター360にセットされるように、0からメモリーブロック310全体の大きさまで占めることができる。

【0028】本発明に従うと、メモリーブロック310におけるメモリーバンクの割り当ては柔軟性を備え、メモリーブロック310におけるいかなる大きさのメモリーバンクがエージェントグループ340、342の何れかに割り当てられることを可能とする。このような構成のため、処理エージェント302〜306、312〜316を動作させる前あるいは構成レジスター360における値を変えるようにコードにおける特定の命令を実行することによってオンザフライ (on-the-fly) でユーザはメモリー構成を変えることができる。

【0029】構成レジスター360はエージェント302、304、306、312、314、316の何れによっても書き込まれることができる。この実施例において、構成レジスター360に書き込まれる値は第1パーティション310aの長さに対応し、残りのメモリーバンクは第2パーティションに割り当てられる。

【0030】勿論、複数のワードないしレジスターを構成レジスター360にて実装することができ、共有メモリーブロック310にて3以上の構成可能パーティションを備えることができる。

【0031】メモリーブロック310は隣接パーティションへとパーティションすることが好ましい (必要ではないが)。例えば、もし構成レジスター360が1の4ビットレジスターからなれば、16までのメモリーバンクの何れの間パーティションを適切に表すことができるようになる。

【0032】例えば、説明の便宜上のため、メモリーブロック310にて14のメモリーバンクがあるとすると。エージェント302、304、306、312、314、316の何れかが360に値「9」(0101H)を書き込めば、このことはアービトラータ312によって、最初の9の個々のメモリーバンクが第1エージェント340に割り当てられ、残りの5の個々のメモリーバンクが第2エージェントグループ342に割り当てられるようにメモリーブロック310を区分するように解釈される。

【0033】次に構成レジスター360の4ビットはア

ービットレーター 312 によってデコードされ、第 1 MUX 308 に適切な制御信号を与えて第 1 エージェントグループ 340 における、要求ないし勝利エージェントを選択したり選択しなかったりし、第 2 MUX 308 に対して、第 2 エージェントグループ 342 における、要求ないし勝利エージェントを選択したり選択しなかったりさせる。メモリーブロック 310 が区分されているので、両方のエージェントグループ 340、342 は衝突なしにそれらの対応する割り当てられたパーティション 310a、310b にアクセスすることができる。

【0034】またメモリーバンクアービトラータは所望のようにメモリースペースに穴を設けることができる。例えば、第 1 エージェントがメモリーバンク 71、2、4、6、8 を割り当てられ、第 2 エージェントがメモリーバンク 14~9、7、5、3 を割り当てられるようにすることができる。

【0035】複数のエージェントシステムにおいて、システムにおける全てのエージェントが用いるための 1 つの共有メモリーブロックがしばしば設けられる。多くの理由によって（例えば、何れかのユーザに対してエージェントそれぞれにおけるアプリケーションプログラムが頻繁に変化すること）、システムにおける各エージェントに対する固定量のメモリーが非効率的となる

【0036】例えば、もしメモリーアクセシングシステムからなる IC において 2 つの利用可能な DSP の内 1 つのみにあって 1 のユーザがコードを実装する場合、そのユーザが第 2 DSP によるメモリー要求があれば最低限しか与えられなくなってしまう。この場合、全ての利用可能な共有メモリーを第 1 DSP に割り当てない区分し、第 2 DSP にメモリーを割り当てないようにすることが好ましいことがある。

【0037】図 2 には、第 1 エージェントバス 496 または第 2 エージェントバス 497 の何れかによってアクセスされるように割り当てることができる複数の個々のメモリーバンク 1~14 からなるメモリーシステム 410 を示している。メモリーバンク 1~14 は同期メモリー（例、SDRAM）あるいは非同期メモリー（例、DRAM）のいずれかでもよい。また、図 2~5B の実施例において 14 の個々のメモリーバンクに関して示してあるが、本発明は任意の数の個々のメモリーバンクを有する共有メモリーシステムにおいて同様に用いることができる

【0038】コモンメモリーシステムのメモリーバンク 1~14 は本発明に従って 2 のエージェントバスによってアクセスされるために柔軟性を持って割り当てられる。この割り当ては、最後から最初、最初から最後のメモリーバンクと逆の方向で行われる。

【0039】個々のメモリーバンク 1~14 はそれぞれ、適切なマルチプレクサ 401~414 を用いることによって何れかのエージェントによって構成可能にアク

セス可能である。MUX 401~414 は選択ロジック 412b によって個別に制御され、これは構成レジスタ 460b にセットされた情報に基づいてメモリーシステム 410 において柔軟性のあるパーティションを有効に実装する。

【0040】この柔軟性のあるパーティションは、メモリーシステムの動作時に移動することができるが、これは全ての影響されるエージェントによって確認されてから行われることが好ましい。従って、1 のエージェントが更にメモリーを必要とすると、そのエージェントがアクセスできるメモリーを増やすことができ、他のエージェントがアクセスできるメモリーは減る。

【0041】本発明は、2 のエージェントの間で複数のメモリーバンクを区分するだけではなく、逆順でメモリーバンクを割り当てる。図 3A~5B は、個々のメモリーバンク 1~14 をエージェント 1（図 3A、4A、5A）とエージェント 2（図 3B、4B、5B）のメモリーマップへと割り当てる実施例を示してある。

【0042】本発明の原理に従い、第 1 メモリーバンクは、図 3A に示すように第 1 エージェントのメモリーマップへと、好ましくは順番に割り当てられる。従って、もしメモリーシステム 410 が第 1 エージェントがアクセスできる 7 のメモリーバンクおよび第 2 エージェントがアクセスできる 7 のメモリーバンクとなるように初期に区分されれば、図 3A に示すように最初の 7 のメモリーバンクが第 1 のエージェントのメモリーマップへと昇順に割り当てられる。これに対し図 3B に示すように、他のエージェントに対するメモリーバンクは、降順（最高から最低へと）で第 2 エージェントのメモリーマップへと最低から最高へと割り当てられる。

【0043】図 4A、B は、5 のメモリーバンクが第 1 エージェントに割り当てられ 9 のメモリーバンクが第 2 エージェントへ割り当てられるような個々のメモリーバンク 1~14 の割り当ての例を示してある。最低のメモリーバンクは最低のメモリーバンク（例えば、メモリーバンク #1）から始まって昇順で第 1 エージェントへ割り当てられ、最高のメモリーバンクは最高のメモリーバンク（例えば、メモリーバンク #14）から始まり昇順で第 2 エージェントに割り当てられる。

【0044】同様に、図 5A、B は、第 1 エージェントが 9 の最低メモリーバンク（順にメモリーバンク #1~#9）を用いるように初期に割り当てられ、第 2 エージェントが 5 の最高メモリーバンク（順にメモリーバンク #14~#10）を用いるように割り当てられたメモリーバンク 1~14 のマッピングの例を示してある。

【0045】本発明は、2 のエージェントから異なる方法でアドレッシングされたメモリーバンクを得る。にもかかわらず、メモリーバンクの間のパーティションの位置を単に変えることによって、第 1 エージェントが記憶するメモリーバンクにおけるデータは他のエージェント

10

20

30

40

50

によってもアクセスすることができる。

【0046】本発明は多くの使用例がある。例えば、2 ケートDSPシステムにおいて、第1DSPが7以上のメモリーバンクを必要とするモデムアプリケーションプログラムを動作させていて、第2DSPが他の7のメモリーバンクのバランスを必要とする中品質のオーディオセッションを動作させているとする。この場合、構成レジスターは、例えば、「7」(0111H)にセットされ、最低から最高へと順に第1の7のメモリーバンク1~7の割り当てを第1DSPに指示し、残りの7のメモリーバンク14~8を最高から最低へと順に割り当てることを第2DSPに指示する。次にその後の時点において、ユーザは、12のメモリーバンクを必要とする第2DSPにて高品質のオーディオセッションを動作させることができる。

【0047】構成レジスター460におけるセッティングを第2DSPへの適切なより多くのメモリーバンクへと第1または第2DSPの何れかが調整することができる。これは、第1DSPのメモリーパーティションの大きさの犠牲を伴う。例えば、この場合、値「2」(0010H)が構成レジスター460に書き込まれ、第1DSPに2のメモリーバンク1、2を割り当て、残りの12のメモリーバンク14~3を第2DSPによって使用するためメモリーブロックの第2パーティションへと構成する。逆に、ユーザは何れの時点においても、大量メモリーを必要とする高ボーレートモデムプログラムを第1DSP上で動作させたいことがある。この場合、構成レジスター460は、例えば、最初の11のメモリーバンク1~11を第1DSPに割り当て、残りの3のメモリーバンク14~12のみを第2DSPの使用のため残すように書き込まれることができる。

【0048】このように本発明に従って、2の処理エージェントがメモリースペースをアドレッシングする方法を変更せずにメモリーアクセスを促進するため調整する必要なしに、隣接メモリースペースにて共有メモリーサブシステムをアクセスすることができる。

【0049】このように、他の処理エージェントの受け取ったメモリースペースを変更せずにメモリーシステムの一部ないし全てを何れかの処理エージェントが利用することができる。

【0050】より一般的には、Mのメモリーバンクを有する共有メモリーバンクシステムにおいて、第1処理エージェントがメモリーバンク1~Nを所有し、第2処理エージェントがメモリーバンクN+1~Mを所有する。第1処理エージェントの観点からは、アドレス0が第1メモリーバンク1に配置され、メモリーバンクの数が増えたとアドレスも増える。第2処理エージェントについては、アドレス0は最高メモリーバンクMiに配置され、メモリーバンクの数が減るとアドレスが増える。これは、隣接メモリー必要条件の場合である。勿論メモリー

バンクの割り当ては隣接条件的でなくてもよい。

【0051】従って、本発明の原理に従うと、2の処理エージェントの間単にデータを交換したり、あるいは調整された処理エージェントの間で別々にデータを交換することができる。例えば、第1処理エージェントによってメモリーバンクにデータを書き込み、次にそのメモリーバンクは、マルチプレクサを適当にスイッチングすることにより第2処理エージェントのドメインへと単純にスイッチングすることができる。このマルチプレクサは、そのメモリーバンクへの入力を制御し、第2処理エージェントに対しデータへのフルアクセスを可能とさせる。

【図面の簡単な説明】

【図1】各アクセスインエージェントへ所定数のメモリーバンクを割り当てるように構造上区分された共通メモリーの図。

【図2】図1に示したパーティショナブルメモリーシステムの実施例の図。

【図3】本発明に従い、2人のエージェントが逆順でアクセスするために割り当てられた複数のメモリーバンクの第1の例を示すメモリー割当図。

【図4】図3と同様なメモリー割当図。

【図5】図3と同様なメモリー割当図。

【図6】別々の非同期メモリーシステムを備える従来の2エージェントシステムを示す図。

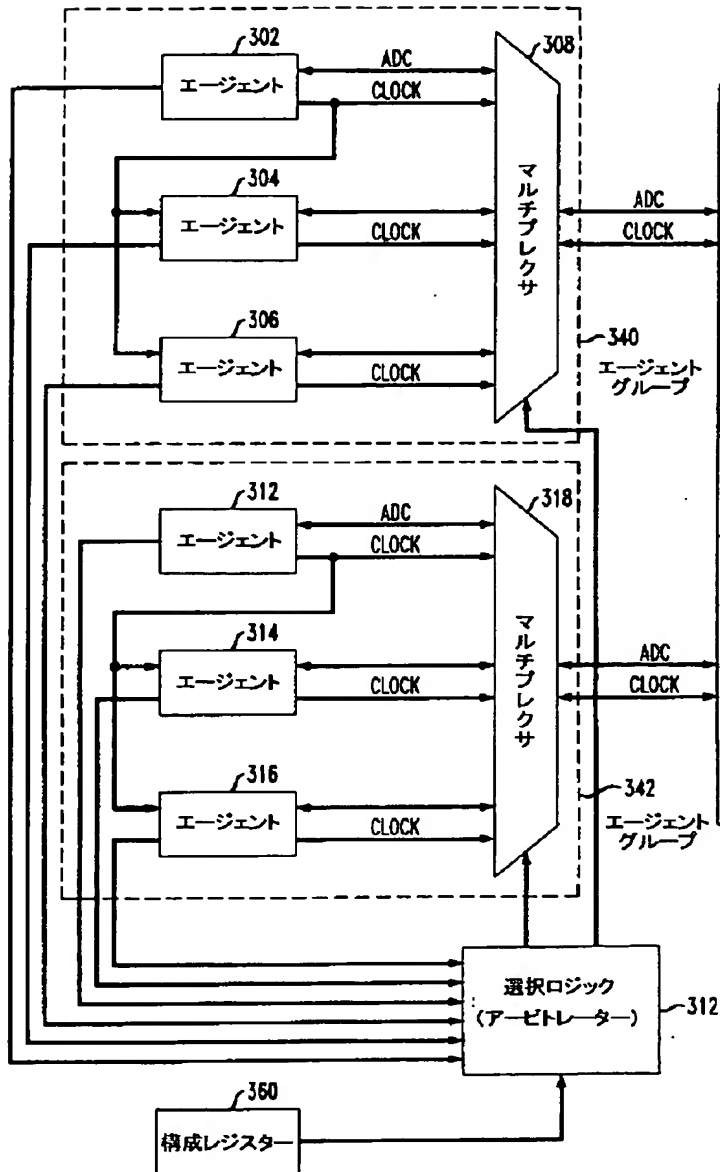
【図7】非同期共有メモリーにアクセスするためアービトラータを用いた従来技術の2エージェントシステムを示す図。

【図8】エージェントの内の1人からのメモリーアクセスをオフセットするため、メモリー管理アドレストランスレータを用いた従来技術のエージェントシステムを示す図。

【符号の説明】

0 アドレス
302~306、312~316、502、504、602、604、702、704 処理エージェント
308、318、401~414、510 マルチプレクサ (MUX)
310 メモリーブロック
310a、310b パーティション
340、342 エージェントグループ
360、460、460b 構成レジスター
1~14 メモリーバンク
410 メモリーシステム
412b 選択ロジック
496、497 エージェントバス
508 シングルポート型コモンメモリーバンクシステム
508a、508b 専用メモリーバンクシステム
512 アービトラータ

【図1】



【図3】

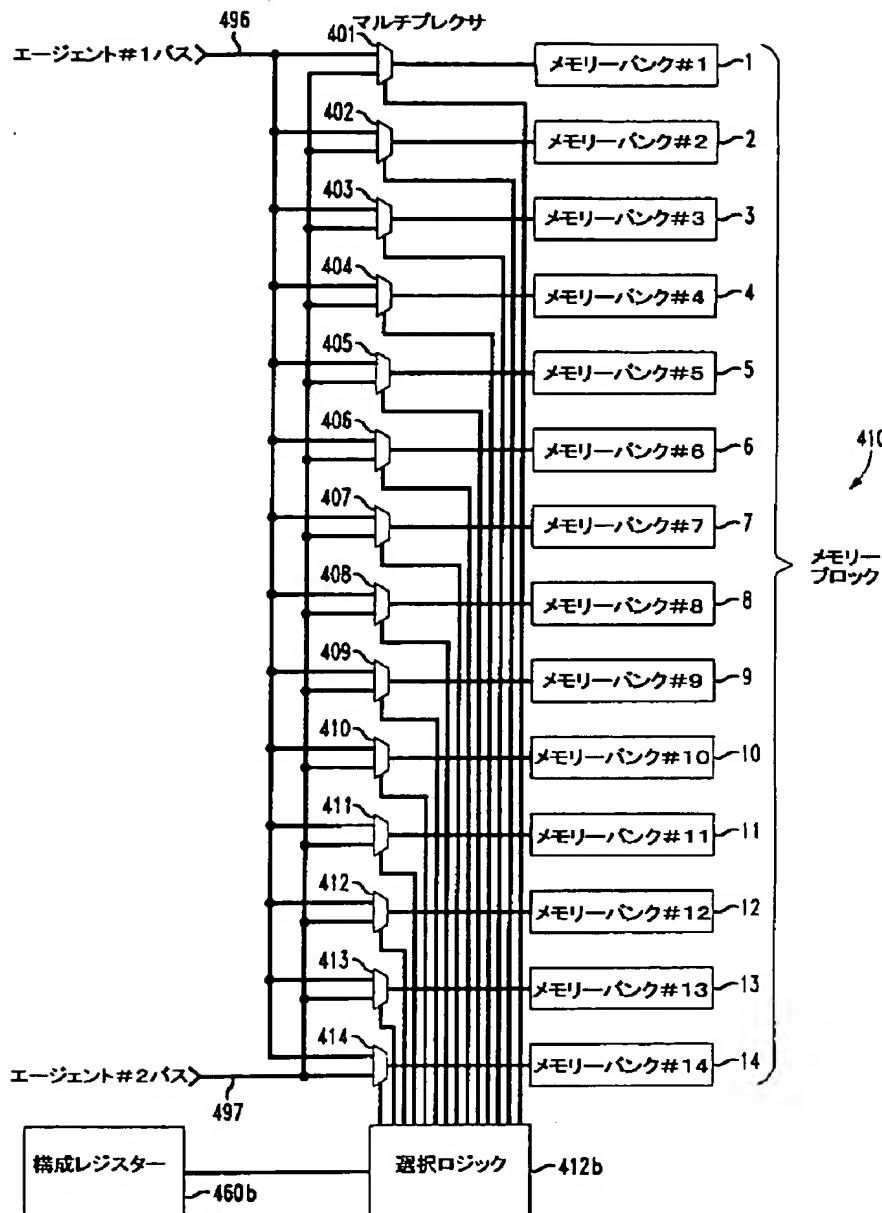
A エージェント#1
メモリーテーブル

7K	メモリーバンク#7
6K	メモリーバンク#6
5K	メモリーバンク#5
4K	メモリーバンク#4
3K	メモリーバンク#3
2K	メモリーバンク#2
1K	メモリーバンク#1
0	

B エージェント#2
メモリーテーブル

7K	メモリーバンク#8
6K	メモリーバンク#9
5K	メモリーバンク#10
4K	メモリーバンク#11
3K	メモリーバンク#12
2K	メモリーバンク#13
1K	メモリーバンク#14
0	

【図2】



【図4】

A エージェント#1
メモリーテーブル

5K	メモリーバンク#5
4K	メモリーバンク#4
3K	メモリーバンク#3
2K	メモリーバンク#2
1K	メモリーバンク#1
0	

B エージェント#2
メモリーテーブル

9K	メモリーバンク#6
8K	メモリーバンク#7
7K	メモリーバンク#8
6K	メモリーバンク#9
5K	メモリーバンク#10
4K	メモリーバンク#11
3K	メモリーバンク#12
2K	メモリーバンク#13
1K	メモリーバンク#14
0	

【図5】

A エージェント#1
メモリーテーブル

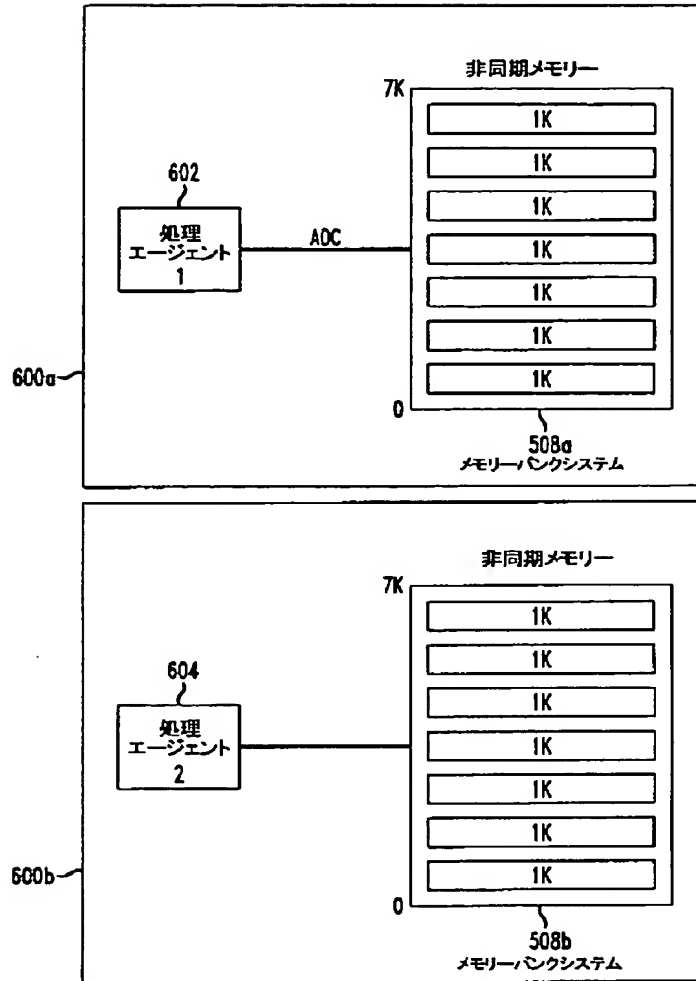
9K	メモリーバンク#9
8K	メモリーバンク#8
7K	メモリーバンク#7
6K	メモリーバンク#6
5K	メモリーバンク#5
4K	メモリーバンク#4
3K	メモリーバンク#3
2K	メモリーバンク#2
1K	メモリーバンク#1
0	

B エージェント#2
メモリーテーブル

5K	メモリーバンク#10
4K	メモリーバンク#11
3K	メモリーバンク#12
2K	メモリーバンク#13
1K	メモリーバンク#14
0	

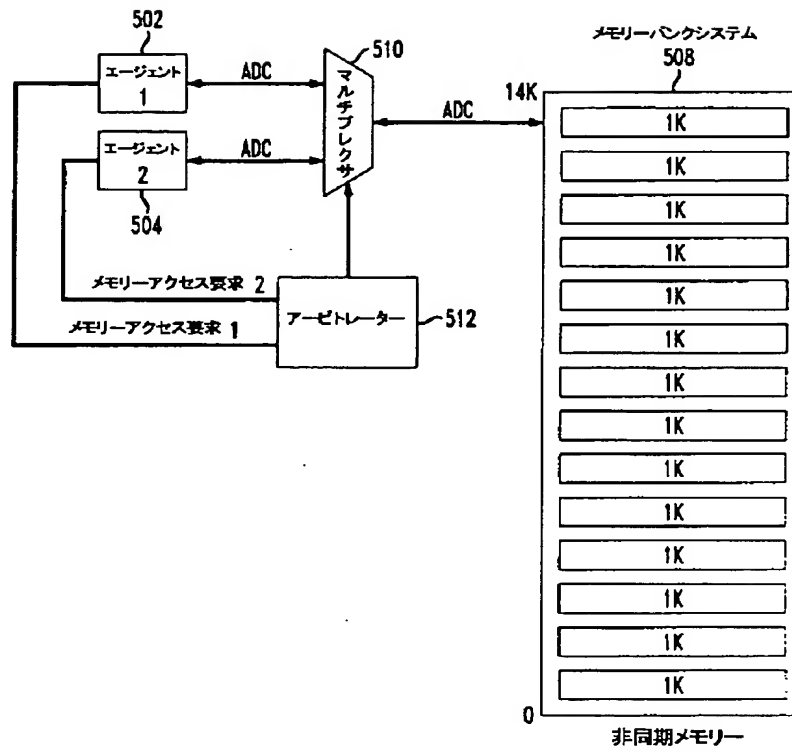
【図6】

従来技術



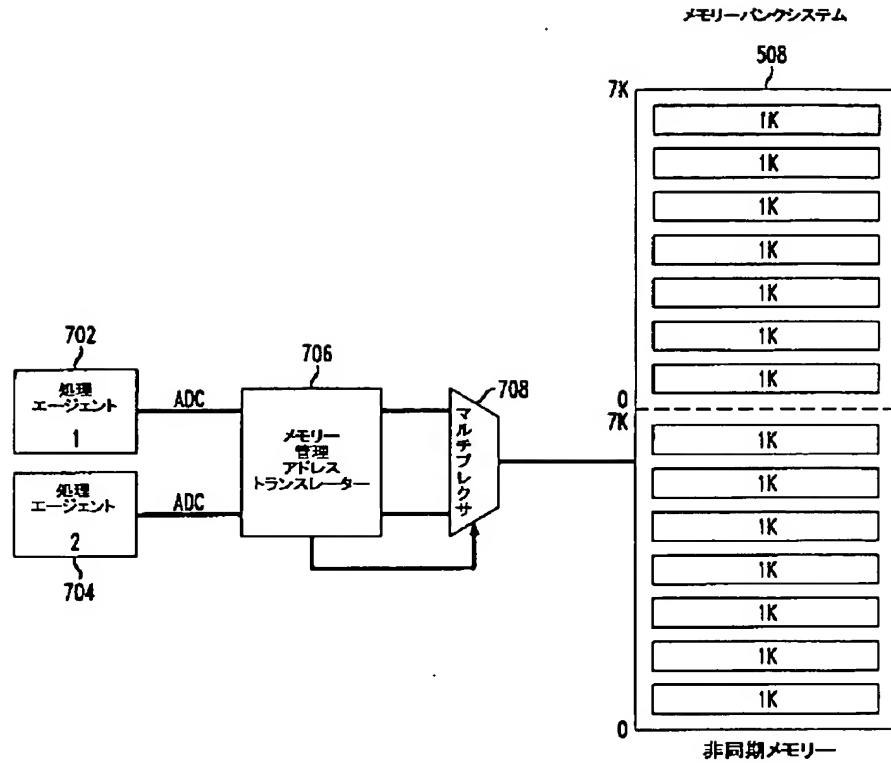
【図7】

従来技術



【図8】

従来技術



フロントページの続き

(71)出願人 596077259
 600 Mountain Avenue,
 Murray Hill, New Je
 rsey 07974-0636 U. S. A.

(72)発明者 リチャード ジョセフ ニーシアー
 アメリカ合衆国, 18015 ペンシルバニア,
 ベスレヘム, ピーチ ツリー レイン
 1937

【外国語明細書】

1. Title of Invention

A Memory System

2. Claims

1. A memory system comprising:

a plurality of memory banks collectively having a lowest addressable end and a highest addressable end;

a control module to assign a first sub-plurality of said memory banks in increasing order from said lowest addressable end to a first memory map of a first processing agent, and to assign a second sub-plurality of said memory banks in decreasing order from said highest addressable end to a second memory map of a second processing agent.

2. The memory system according to claim 1, wherein:

said plurality of memory banks are contiguous.

3. The memory system according to claim 1, further comprising:

a switch adapted and arranged to switch each of said plurality of memory banks for access from one of said first processing agent and said second processing agent.

4. The memory system according to claim 1, wherein said control module comprises:

a configuration register to define said first sub-plurality of said memory banks and said second sub-plurality of said memory banks.

5. The memory system according to claim 1, wherein:

said plurality of memory banks are assigned to only one of said first processing agent and said second processing agent at any one time.

6. The memory system according to claim 1, wherein:

said plurality of memory banks are synchronous memory banks.

7. The memory system according to claim 1, wherein:

said plurality of memory banks are asynchronous memory banks.

8. A method of assigning a plurality of memory banks for shared use among a plurality of processing agents, said method comprising:

providing a first processing agent access to a plurality of memory banks with a lower address of said first processing agent being associated with a last one of said plurality of memory banks and a higher address of said first processing agent being associated with a last one of said plurality of memory banks; and

providing a second processing agent access to said plurality of memory banks with a lower address of said second processing agent being associated with said last one of said plurality of memory banks and a higher address of said second processing agent being associated with a first one of said plurality of memory banks.

9. The method of assigning a plurality of memory banks for shared use among a plurality of processing agents according to claim 8, wherein:

said first one of said plurality of processing agents is associated with address zero in a memory map of said first processing agent.

10. The method of assigning a plurality of memory banks for shared use among a plurality of processing agents according to claim 8, wherein:

said last one of said plurality of processing agents is associated with address zero in a memory map of said second processing agent.

11. The method of assigning a plurality of memory banks for shared use among a plurality of processing agents according to claim 9, wherein:

said last one of said plurality of processing agents is associated with address zero in a memory map of said second processing agent.

12. The method of assigning a plurality of memory banks for shared use among a plurality of processing agents according to claim 9, wherein:

said plurality of memory banks are synchronous memory.

13. The method of assigning a plurality of memory banks for shared use among a plurality of processing agents according to claim 9, wherein:

said plurality of memory banks are asynchronous memory.

14. The method of assigning a plurality of memory banks for shared use among a plurality of processing agents according to claim 9, wherein:

each of said plurality of memory banks are not substantially simultaneously accessible by both said first processing agent and said second processing agent at a same time.

15. A method of assigning a plurality of shared memory banks to two agents, comprising:

assigning a plurality of shared memory banks in increasing order to a memory map of a first agent; and

assigning said plurality of shared memory banks in decreasing order to a memory map of a second agent.

16. The method of assigning a plurality of shared memory banks to two agents according to claim 15, wherein:

said plurality of shared memory banks are synchronous memory banks.

17. A method of assigning a plurality of shared memory banks to two agents, comprising:

assigning a plurality of shared memory banks in opposing order to each of two processing agents.

18. Apparatus for assigning a plurality of memory banks for shared use among a plurality of processing agents, comprising:

means for providing a first processing agent access to a plurality of memory banks with a lower address of said first processing agent being associated with a last one of said plurality of memory banks and a higher address of said first processing agent being associated with a last one of said plurality of memory banks; and

means for providing a second processing agent access to said plurality of memory banks with a lower address of said second processing agent being associated with said last one of said plurality of memory banks and a higher address of said second processing agent being associated with a first one of said plurality of memory banks.

19. The apparatus for assigning a plurality of memory banks for shared use among a plurality of processing agents according to claim 18, wherein:

said first one of said plurality of processing agents is associated with address zero in a memory map of said first processing agent.

20. The apparatus for assigning a plurality of memory banks for shared use among a plurality of processing agents according to claim 18, wherein:

said last one of said plurality of processing agents is associated with address zero in a memory map of said second processing agent.

21. The apparatus for assigning a plurality of memory banks for shared use among a plurality of processing agents according to claim 20, wherein:

said last one of said plurality of processing agents is associated with address zero in a memory map of said second processing agent.

22. The apparatus for assigning a plurality of memory banks for shared use among a plurality of processing agents according to claim 18, wherein:

said memory is synchronous memory.

23. Apparatus for assigning a plurality of shared memory banks to two agents, comprising:

means for assigning a plurality of shared memory banks in increasing order to a memory map of a first agent; and

means for assigning said plurality of shared memory banks in decreasing order to a memory map of a second agent.

--

24. The apparatus for assigning a plurality of shared memory banks to two agents according to claim 23, wherein:

said plurality of shared memory banks are synchronous memory banks.

25. The apparatus for assigning a plurality of shared memory banks to two agents, comprising:

means for assigning a plurality of shared memory banks in opposing order to each of two processing agents.

26. The apparatus for assigning a plurality of shared memory banks to two agents according to claim 25, wherein:

said plurality of shared memory banks comprise synchronous memory.

27. A computer system, comprising:

a first processing agent;

a second processing agent;

a shared synchronous memory accessible by said first processing agent and by said second processing agent; and

a control module to assign a first portion of said shared synchronous memory in increasing order from a lowest addressable end to a first memory map of said first processing agent, and to assign a second sub-plurality of said shared synchronous memory in decreasing order from a highest addressable end to a second memory map of said second processing agent.

3. Detailed Description of Invention

Field of the Invention

This invention relates generally to the shared usage of memory by a plurality of processing agents. In particular, it relates to the efficient and flexible partitioning of shared single-ported memory between a plurality of processing agents.

Background of Related Art

With the ever-increasing speeds of today's processors, memory designs have attempted to meet the required speed requirements. Moreover, with the ever-increasing integration of more than one specific task oriented processor for use in a single system, the efficient utilization of fast memory has become a design challenge.

For instance, to achieve fast memory speeds, synchronous dynamic random access memory (SDRAM) technology has been developed for a wide variety of applications to close the gap between the needs of high-speed processors and the access time of non-synchronous memory such as dynamic random access memory (DRAM) or static random access memory (SRAM). Synchronous memory, e.g., SDRAM technology, combines industry advances in fast dynamic random access memory (DRAM) with a high-speed interface.

Functionally, an SDRAM resembles a conventional DRAM, i.e., it is dynamic and must be refreshed. However, the SDRAM architecture has improvements over standard DRAMs. For instance, an SDRAM uses internal pipelining to improve throughput and on-chip interleaving between separate memory banks to eliminate gaps in output data.

The idea of using a SDRAM synchronously (as opposed to using a DRAM asynchronously) emerged in light of increasing data transfer demands of high-end processors. SDRAM circuit designs are based on state machine operation instead of being level/pulse width driven as in conventional asynchronous memory devices. In addition, synchronous memory access techniques improve the margin to system noise because inputs are not level driven. Instead, the inputs are latched by the system clock. Since all timing is based on the same synchronous clock, designers can achieve better specification margins. Moreover, since the SDRAM access is programmable, designers can improve bus utilization because the processor can be synchronized to the SDRAM output.

The core of an SDRAM device is a standard DRAM with the important addition of synchronous control logic. By synchronizing all address, data and control signals with a single clock signal, SDRAM technology enhances performance, simplifies design and provides faster data transfer.

Synchronous memory requires a clock signal from the accessing agent to allow fully synchronous operation with respect to the accessing agent. If more than one agent is given access to a shared synchronous memory, each agent must conventionally supply its own clock signal to the synchronous memory. Unfortunately, the clock signals from separate agents are not conventionally synchronous or in phase with one another. Therefore, as the synchronous memory shifts from the use of one clock signal to another, delays

or wait states must be added before the new agent's clock signal can be used to access the synchronous memory.

Some synchronous memory devices have the capability to provide burst input/output (I/O), particularly for the optimization of cache memory fills at the system frequency. Advanced features such as programmable burst mode and burst length improve memory system performance and flexibility in conventional synchronous memories, and eliminate the need to insert wait states, e.g., dormant clock cycles, between individual accesses in the burst.

Conventional SDRAM devices include independent, fixed memory sections that can be accessed individually or in an interleaved fashion. For instance, two independent banks in an SDRAM device allow that device to have two different rows active at the same time. This means that data can be read from or written to one bank while the other bank is being precharged. The setup normally associated with precharging and activating a row can be hidden by interleaving the bank accesses.

Conventionally, each task oriented processing agent had been provided with its own dedicated memory bank system, e.g., as shown in Fig. 6. Fig. 6 shows two separate processing agent systems 600a, 600b each having a processing agent 602, 604 and a dedicated memory bank system 508a, 508b, respectively. Each processing agent 602, 604 may be a suitable processor, e.g., a microprocessor, a microcontroller, or a digital signal processor (DSP).

Unfortunately, for applications wherein one processing agent requires significant amounts of information from another processing agent, significant delays were caused in transferring the information between the memory bank system of the first processing agent and the memory bank system of the second processing agent to allow the second processing agent to

co-process information along with the first processing agent. To reduce the need for transferring information between non-synchronous memory bank systems, shared use of conventional non-synchronous memory using an arbitrator was developed, e.g., as shown in Fig. 7.

In particular, Fig. 7 shows a common single-ported memory bank system 508 being accessed by two processing agents 502, 504. A multiplexer (MUX) 510 passes the address, data and control busses (ADC) of a selected processing agent 502, 504 to the common single-ported memory bank system 508 under the control of an arbiter or arbitrator 512.

As may be appreciated, memory accesses by the separate agents 502, 504 would clash unless they are arbitrated to allow only one agent to access the common single-ported memory bank system 508 at any one time. Thus, selection logic (i.e., an arbitrator 512) is conventionally provided to control the multiplexer 510, which presents the appropriate address, data and control (ADC) signals to the common single-ported memory bank system 508. Typically, the processing agents 502, 504 are assigned a hierarchy in the arbitrator 512 for prioritized access to the common single-ported memory bank system 508, and the arbitrator 512 blocks out accesses by the other agents until finished accessing the common single-ported memory bank system 508.

Another conventional technique for sharing common single-ported non-asynchronous memory utilizes a memory management address translator 706 as shown in Fig. 8.

In particular, Fig. 8 shows two processing agents 702, 704 accessing a common single-ported memory bank system 508 under the control of a memory management address translator 706 and MUX 708. The address translation function serves to invisibly index the memory accesses by one of the processing agents 702, 704 up by a predetermined amount of memory,

e.g., 7K as shown in Fig. 8. Thus, for instance, memory accesses to address 0 by the first processing agent 702 will index through the memory management address translator to a physical address 0 in the common single-ported memory bank system 508, and memory accesses to address 0 by the second processing agent 704 will index through the memory management address translator to an offset physical address, e.g., 7K in the common single-ported memory bank system 508.

Unfortunately, address translation is conventionally fixed in shared memory systems and does not provide flexibility in the changing memory needs of the processing agents, particularly as the agents are in operation.

Thus, there is a need for memory systems in general which adjust to the changing needs of a plurality of accessing agents.

Summary Of The Invention

In accordance with the principles of the present invention, a memory system comprises a plurality of contiguous memory banks collectively having a lowest addressable end and a highest addressable end. A switch is adapted and arranged to switch each of the plurality of contiguous memory banks for access from one of a first processing agent and a second processing agent. A control circuit assigns a first sub-plurality of the contiguous memory banks in increasing order from the lowest addressable end to a first memory map of the first processing agent, and assigns a second sub-plurality of the contiguous memory banks in decreasing order from the highest addressable end to a second memory map of the second processing agent.

A method of assigning a plurality of memory banks for shared use among a plurality of processing agents in accordance with the principles of the present invention includes providing a first processing agent access to a

plurality of memory banks with a lower address of the first processing agent being associated with a last one of the plurality of memory banks and a higher address of the first processing agent being associated with a last one of the plurality of memory banks. A second processing agent is provided access to the plurality of memory banks with a lower address of the second processing agent being associated with the last one of the plurality of memory banks and a higher address of the second processing agent being associated with a first one of the plurality of memory banks.

In another aspect of the present invention, a method of assigning a plurality of shared memory banks to two agents comprises assigning a plurality of shared memory banks in increasing order to a memory map of a first agent. The plurality of shared memory banks are also assigned in decreasing order to a memory map of a second agent.

In yet another aspect of the present invention, a method of assigning a plurality of shared memory banks to two agents comprises assigning a plurality of shared memory banks in opposing order to each of two processing agents.

Detailed Description Of Illustrative Embodiments

Figs. 1 and 2 show a single-ported memory system including a plurality of memory blocks accessible by two separate agents or groups of processing agents. The present invention relates to the assignment of a plurality of shared memory blocks between two agents.

In particular, Fig. 1 shows a shared memory configurably partitioned to assign a pre-defined number of memory banks in a common memory system 310 to each accessing agent or group of agents. Although Fig. 1 shows synchronous memory 310, the present invention is also applicable to

the use of shared non-synchronous memory, i.e., conventional memory such as dynamic random access memory (DRAM) or static random access memory (SRAM).

The embodiment of Fig. 1 is shown with respect to two groups of accessing agents, i.e., a first group of shared memory accessing agents 340 including processing agents 302-306, and a second group of shared memory accessing agents 342 including processing agents 312-316. Although the agent groups 340, 342 are shown with an equal number of agents, the present invention is equally applicable to groups of accessing agents having any number of agents.

The memory block 310 is configurably partitioned into a corresponding number of partitions 310a, 310b. Although the embodiment is shown with two partitions 310a, 310b, the invention is equally applicable to any number of partitions. Preferably, the number of partitions will equal the number of agent groups.

The separate agent groups 340, 342 access respectively partitioned portions of the shared memory 310. The partitions 310a, 310b may be any size, from zero to the entire memory block 310, as set in a configuration register 360.

In accordance with the principles of the present invention, the assignment of the memory banks in the memory block 310 is flexible, allowing any or all of the memory banks in the memory block 310 to be assigned to either of the agent groups 340, 342. With the configurability as disclosed, the user can change the memory configuration either before operating the processing agents 302-306, 312-316 or on-the-fly by executing certain instructions in the code to change the value in the configuration register 360.

The configuration register 360 can be written to by any of the agents 302, 304, 306, 312, 314 or 316. In the disclosed embodiment, the value written to the configuration register 360 corresponds to the length of the first partition 310a, with the remaining memory banks being assigned to the second partition.

Of course, multiple words or registers may be implemented within the configuration register 360 to accommodate more than two configurable partitions in the shared memory block 310 in accordance with the principles of the present invention.

It is preferred (but not required) that the memory block 310 be partitionable into contiguous partitions. For instance, if the configuration register 360 comprises one, four-bit register, it may adequately represent partitions between any of up to sixteen memory banks.

For example, assume for the purpose of explanation that there are fourteen memory banks in the memory block 310. If any of the agents 302, 304, 306, 312, 314 or 316 writes the value '9' (0101H) to the configuration register 360, this would be interpreted by the arbitrator 312 to partition the memory block 310 such that the first nine individual memory banks are to be assigned to the first agent group 340, and that the remaining five individual memory banks are to be assigned to the second agent group 342. The four bits of the configuration register 360 are then decoded by the arbitrator 312 to provide appropriate control signals to a first MUX 308 to select or deselect the requesting or winning agent in the first agent group 340 and to the second MUX 318 to select or deselect the requesting or winning agent in the second agent group 342. Because the memory block 310 is partitioned, both agent groups 340, 342 may access their respectively assigned partition 310a, 310b without conflict.

The memory bank arbitrator can also leave holes in the memory space as desired. For example, a first agent may be assigned memory banks 1, 2, 4, 6 and 8, while a second agent may be assigned memory banks 14-9, 7, 5 and 3.

In a multiple agent system, one shared memory block is often provided for use by all agents in the system. For various reasons, e.g., vastly differing application programs in each of the agents for any one user, a fixed amount of memory for each agent in the system may be inefficient. For instance, if one user implements code in only one of two available DSPs in an integrated circuit comprising a memory accessing system in accordance with the aspects of the present invention, then that user will have minimal if any memory requirement for the second DSP. In that case, it might be desirable to assign or partition all available shared memory to the first DSP and no memory to the second DSP.

Fig. 2 shows a memory system 410 comprising a plurality of individual memory banks 1 to 14 which can be assigned for access either by a first agent bus 496 or a second agent bus 497. The memory banks 1 to 14 may be either synchronous (e.g., SDRAM) memory or non-synchronous (e.g., DRAM) memory. Moreover, although the embodiments of Figs. 2 to 5B are shown with respect to fourteen individual memory banks, the invention is equally applicable to shared memory systems having any number of individual memory banks.

In accordance with the principles of the present invention, the memory banks 1 to 14 of a common memory system are flexibly assigned for access by two agent busses. The assignments are made in opposing directions from last to first and first to last memory bank, respectively.

Each individual memory bank 1 to 14 is configurably accessible by either agent through the use of suitable multiplexers 401 to 414. The MUXs 401 to 414 are individually controlled by selecting logic 412b, which effectively implements a flexible partition in the memory system 410 based on information set in a configuration register 460b.

The flexible partition may be moved during the operation of the memory system, but is preferably done so with confirmation by all affected agents. Thus, as one agent requires additional memory, the accessible memory for that agent can be increased, albeit at the expense of the available memory for the other agent.

Importantly, the present invention not only partitions the plurality of memory banks among the two agents, but assigns the memory banks in opposing order. Figs. 3A through 5B depict exemplary assignments of the individual memory banks 1 to 14 into the memory map of the agent 1 (Figs. 3A, 4A and 5A) and agent 2 (Figs. 3B, 4B and 5B).

In accordance with the principles of the present invention, the first memory banks are assigned, preferably in order, into the memory map of the first agent as shown in Fig. 3A. Thus, if the memory system 410 is to be initially partitioned with seven memory banks accessible by the first agent and seven memory banks accessible by the second agent, then the first seven memory banks are assigned in increasing order into the memory map of the first agent as shown in Fig. 3A. However, as shown in Fig. 3B, the memory banks for the other agent are assigned in decreasing order, i.e., from highest to lowest, into the memory map of the second agent, from lowest to highest.

To further illustrate the invention, Figs. 4A and 4B show an example assignment of the individual memory banks 1 to 14 with five memory banks assigned to the first agent and nine memory banks assigned to the

second agent. In accordance with the principles of the present invention, the lowest memory banks are assigned to the first agent in increasing order starting with the lowest memory bank, e.g., memory bank #1, while the highest memory banks are assigned to the second agent in decreasing order starting with the highest memory bank, e.g., memory bank #14.

Similarly, Figs. 5A and 5B illustrate an example mapping of the memory banks 1 to 14 wherein the first agent is initially assigned use of the nine lowest memory banks, i.e., memory banks #1 to #9 in order, while the second agent is assigned use of the five highest memory banks, i.e., memory banks #14 to #10 in order.

Accordingly, the present invention results in memory banks being addressed differently from the two agents. Nevertheless, by simply changing the position of the partition between the memory banks, data stored in a memory bank by a first agent can be accessed by the other agent.

The present invention has numerous applications. For instance, in a two DSP system, the first DSP may be running a modem application program requiring at least seven memory banks while the second DSP may be running a medium quality audio session requiring the balance of another seven memory banks. In this case, the configuration register may be set to, e.g., a '7' (0111H) to indicate the assignment of the first seven memory banks 1-7 in order from lowest to highest to the first DSP and the remaining seven memory banks 14-8 in order from highest to lowest to the second DSP. Then, at a later point in time, the user may run a high quality audio session at the second DSP which requires 12 memory banks. Either the first or second DSP can adjust the setting in the configuration register 460 to appropriate more memory banks to the second DSP, albeit at the expense of the size of the memory partition for the first DSP. For instance, in this case, a value of '2' (0010H) may be written to the configuration register 460 to assign two memory banks 1, 2 to the first

DSP leaving the remaining twelve memory banks 14-3 in the second partition of the memory block for use by the second DSP. Conversely, the user may at another point in time wish to run a higher baud rate modem program on the first DSP requiring a larger amount of memory. In this case, the configuration register may be written to, e.g., to assign the first eleven memory banks 1-11 to the first DSP leaving only the remaining three memory banks 14-12 for use by the second DSP.

Thus, according to the principles of the present invention, two processing agents can access a shared-memory subsystem within a contiguous memory space without modifying the way in which the two processing agents address that memory space, i.e., memory management block, and without the need to arbitrate to gain access to the memory.

Accordingly, either processing agent can use a portion or all of the memory system without modifying the perceived memory space of the other processing agent.

In more general terms, in a shared memory bank system having M memory banks, a first processing agent will own memory banks 1 through N and a second processing agent will own memory banks N+1 to M. From the perspective of the first processing agent, address zero is located in the first memory bank 1, and the addresses increase with increasing memory bank numbers. From the perspective of the second processing agent, address zero is located in the highest memory bank M and its addresses increase with decreasing memory bank numbers. This is in the case of a contiguous memory requirement. Of course, the memory bank assignments do not necessarily have to be contiguous.

Therefore, in accordance with the principles of the present invention, data can be exchanged simply between two processing agents or separately arbitrated groups of processing agents. For instance, data can be

written into a memory bank by a first processing agent, and then that memory bank can be simply switched into the domain of the second processing agent by appropriate switching of multiplexers controlling the input to that memory bank to allow the second processing agent full access to the data.

4. Brief Description of Drawings

Fig. 1 shows a shared memory configurably partitioned to assign a pre-defined number of memory banks to each accessing agent.

Fig. 2 shows one embodiment of the partitionable memory system shown in Fig. 1.

Figs. 3A and 3B show a first example of a plurality of memory banks assigned for access by two agents in opposing order in accordance with the principles of the present invention.

Figs. 4A and 4B show a second example of a plurality of memory banks assigned for access by two agents in opposing order in accordance with the principles of the present invention.

Figs. 5A and 5B show a third example of a plurality of memory banks assigned for access by two agents in opposing order in accordance with the principles of the present invention.

Fig. 6 shows a conventional two agent system having separate non-synchronous memory systems.

Fig. 7 shows a conventional two agent system utilizing an arbitrator for access to a shared non-synchronous memory.

Fig. 8 shows a conventional two agent system utilizing a memory management address translator to offset memory accesses from one of the agents.

FIG. 1

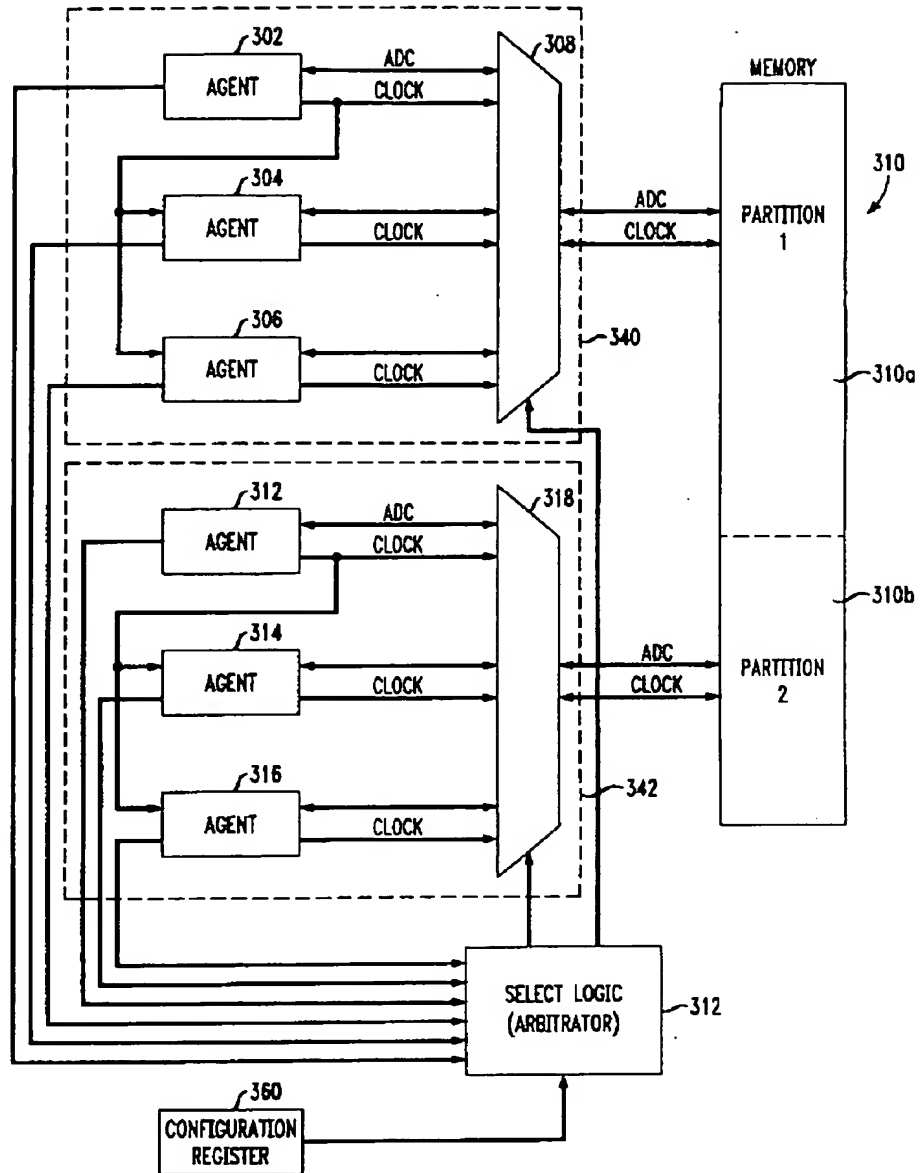


FIG. 2

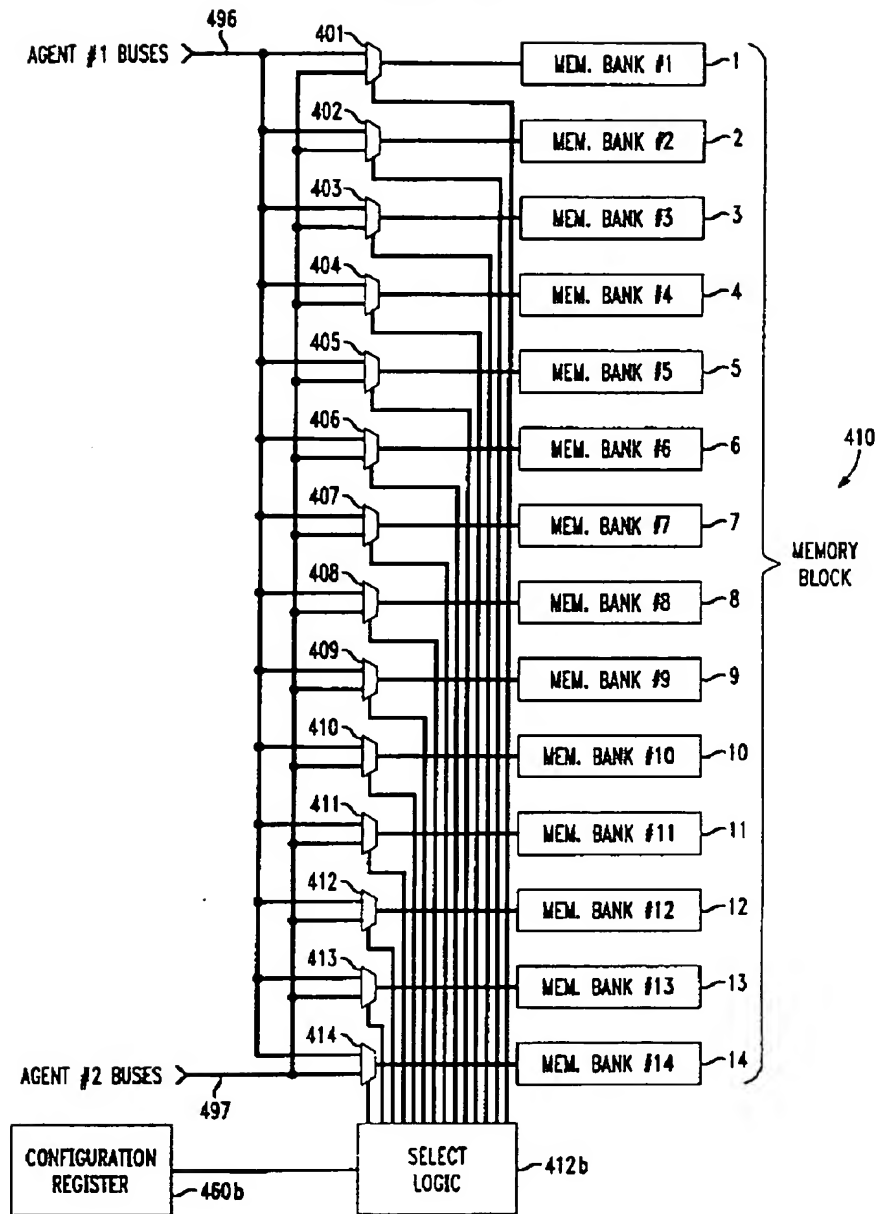


FIG. 3A

AGENT #1
MEMORY TABLE

7K	MEM. BANK #7
6K	MEM. BANK #6
5K	MEM. BANK #5
4K	MEM. BANK #4
3K	MEM. BANK #3
2K	MEM. BANK #2
1K	MEM. BANK #1
0	

FIG. 3B

AGENT #2
MEMORY TABLE

7K	MEM. BANK #8
6K	MEM. BANK #9
5K	MEM. BANK #10
4K	MEM. BANK #11
3K	MEM. BANK #12
2K	MEM. BANK #13
1K	MEM. BANK #14
0	

*FIG. 4A*AGENT #1
MEMORY TABLE

5K	MEM. BANK #5
4K	MEM. BANK #4
3K	MEM. BANK #3
2K	MEM. BANK #2
1K	MEM. BANK #1
0	

*FIG. 4B*AGENT #2
MEMORY TABLE

9K	MEM. BANK #6
8K	MEM. BANK #7
7K	MEM. BANK #8
6K	MEM. BANK #9
5K	MEM. BANK #10
4K	MEM. BANK #11
3K	MEM. BANK #12
2K	MEM. BANK #13
1K	MEM. BANK #14
0	

*FIG. 5A*AGENT #1
MEMORY TABLE

9K	MEM. BANK #9
8K	MEM. BANK #8
7K	MEM. BANK #7
6K	MEM. BANK #6
5K	MEM. BANK #5
4K	MEM. BANK #4
3K	MEM. BANK #3
2K	MEM. BANK #2
1K	MEM. BANK #1
0	

*FIG. 5B*AGENT #2
MEMORY TABLE

5K	MEM. BANK #10
4K	MEM. BANK #11
3K	MEM. BANK #12
2K	MEM. BANK #13
1K	MEM. BANK #14
0	

FIG. 6
PRIOR ART

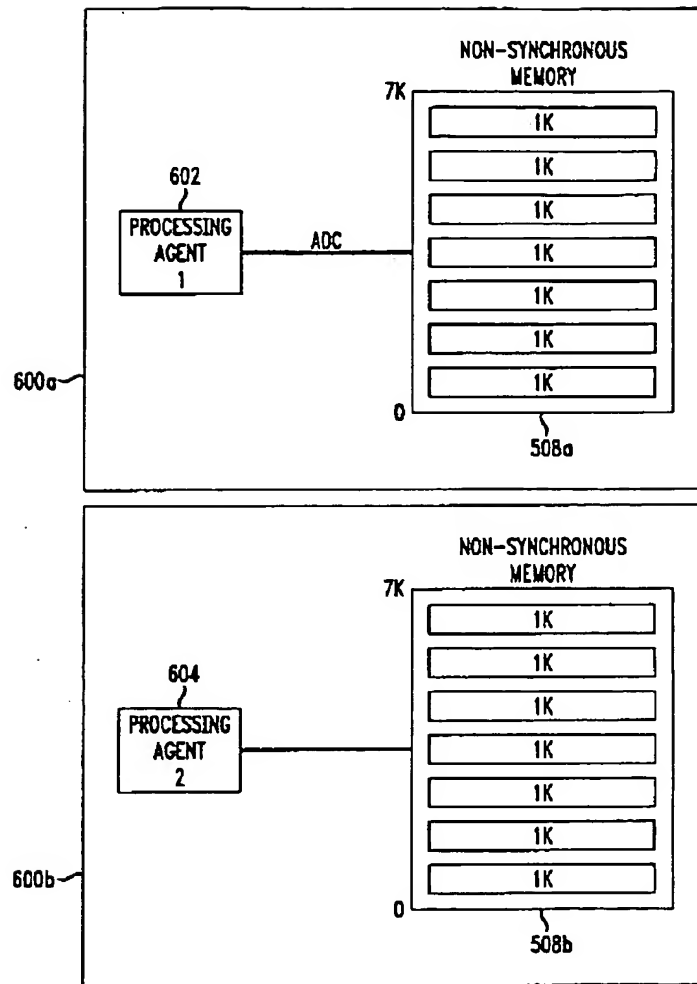


FIG. 7
PRIOR ART

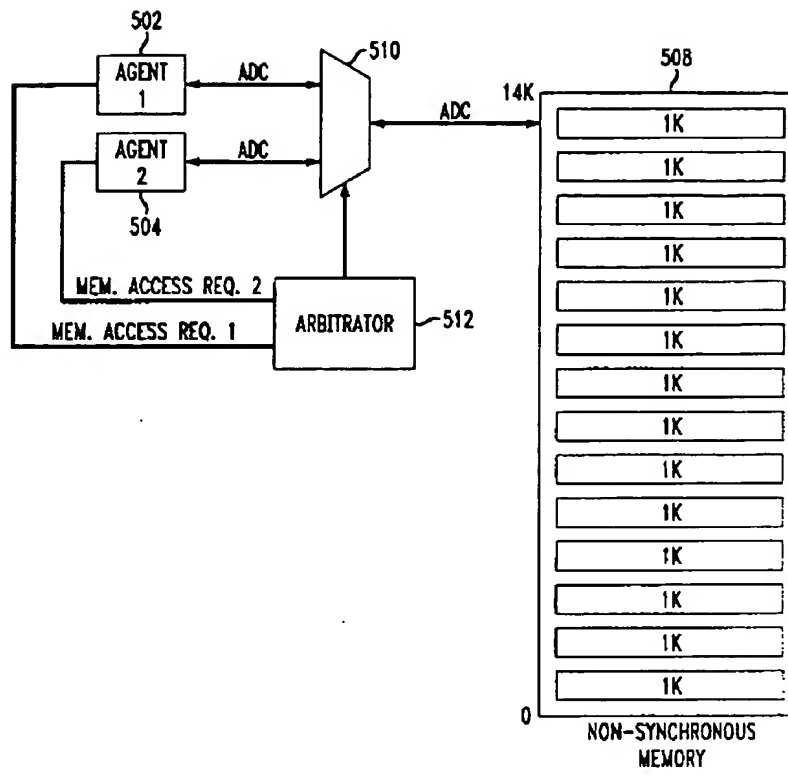
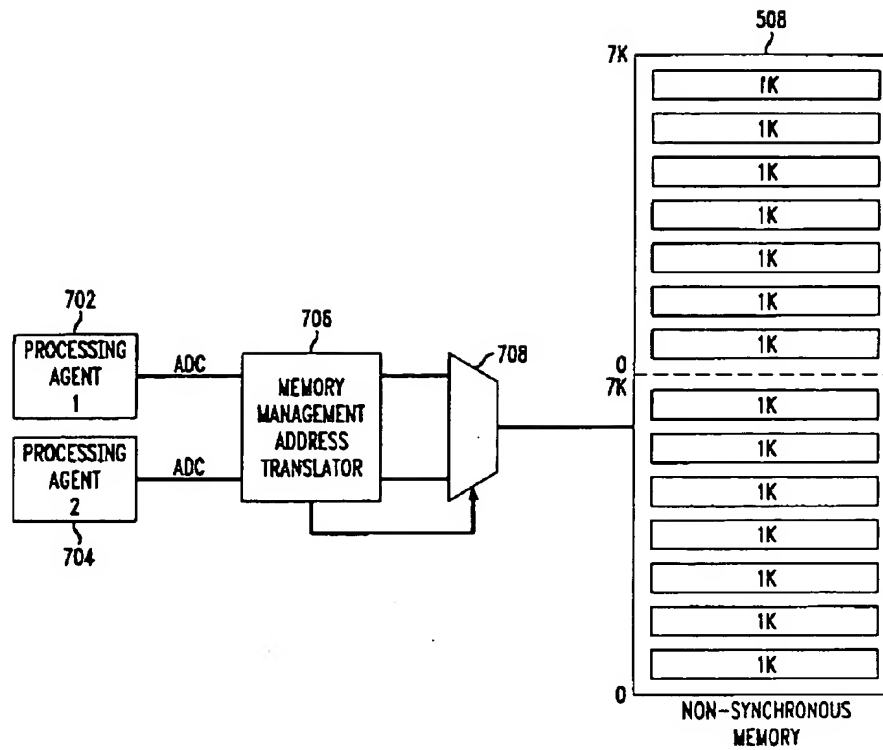


FIG. 8
PRIOR ART



1 Abstract

A shared memory system configurably partitioned to assign a pre-defined number of memory banks to a first processing agent and the remaining memory banks to a second processing agent. The first group of memory banks is assigned into the memory map of the first processing agent in increasing order while the second group of memory banks is assigned into the memory map of the second processing agent in decreasing order. In more general terms, in a shared memory bank system having M memory banks, a first processing agent will own memory banks 1 through N starting at its address 0, and a second processing agent will own memory banks M to $N+1$, respectively, starting at its address 0.

2 Representative Drawing

Figure 1